UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/756,894	01/13/2004	John Randall West	07464-0002	3627

25213 7590 06/07/2007

HELLER EHRMAN LLP
275 MIDDLEFIELD ROAD
MENLO PARK, CA 94025-3506

EXAMINER	
DAO, THUY CHAN	
ART UNIT	**PAPER NUMBER**
2192	

MAIL DATE	DELIVERY MODE
06/07/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/756,894	WEST ET AL.	
	Examiner	Art Unit	
	Thuy Dao	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _03_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 April 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-49* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-49* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 January 2004* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the amendment filed on April 3, 2007.

2. Claims 1-49 have been examined.

Response to Amendments

3. Per Applicants' request, claims 1-2, 14-15, 38, 43, and 49 have been amended.

4. The objection to claims 2 and 49 is withdrawn in view of Applicants' amendments.

5. The 35 USC §112, second paragraph rejection over claims 14 and 15 is withdrawn in view of Applicants' amendments.

6. The 35 USC §101 rejection over claims 1-48 is withdrawn in view of Applicants' amendments. However, the 35 USC §101 rejection over claim 49 is maintained as addressed below.

Response to Arguments

7. The Applicants are thanked for a thorough reply. Applicants' arguments have been fully considered. However, they are either not persuasive (claim 49) or moot in view of the new grounds of rejection (claims 1-48).

Claim 49:

As set forth in the previous Office Action mailed October 6, 2006 (pp. 6-7), Rosaria discloses *converting test cases to an abstract representation that includes application state, external interaction sequences and input data* (e.g., FIG. 3, col.6: 18-65; FIG. 6, col.9: 16 – col.10: 56).

Rosaria further discloses storing data of test case models, intermediate and executable files of the test case generator program in volatile and nonvolatile memories of the system (e.g., FIG. 4, col.7: 32 – col.8: 46), saving graph traversal programs (e.g., "`graph.exe`"), model files (e.g., "`clock.std`"), test case files (e.g., "`clock.tsc`") in database (col.6: 28-37 and col.14: 62 – col.15: 21). Rosaria also discloses a test oracle function can be called to verify results/states/status from the tests stored in the system (e.g., col.4: 16-62).

Thus, it would have been obvious to one having ordinary skill in the art of test case models to store *program instructions of abstract representations of test cases in a database* in order to provide better data organization and manageability for the program application.

Accordingly, Applicants' arguments are not persuasive. The examiner respectfully maintains ground of rejection over claim 49.

Drawings

8. The drawings are objected to because minor informalities:

Figure 2: hand-written reference number 32. Furthermore, in Figure 2, 32 indicates "Sequenced Interactions", wherein in the specification, [0036], 32 indicates "flow control structures". For consistency between Figure 2 and the specification, either "Sequenced Interactions" or "Flow Control Structures" is used for the reference number 32;

Figure 3: hand-written reference number 56. Furthermore, the specification, [0043], sets forth "an identification field 56 within the application metadata repository 54" (emphasis added). However, the newly added identification filed 56 in the Figure 4 is independent/separate with the application metadata repository 54 (i.e., not within the repository 54). For consistency, "56" should be deleted in both Figure 4 and specification;

Figure 6: "54" should be deleted because this reference number has been used to indicate the application metadata repository 54 in Figure 4;

Figure 5: blocks "Read Input Scripts", "Parse the Scripts (Syntax Analysis)", and "Convert To Internal Format (Semantic Analysis)" have two outgoing arrows for every block. The examiner respectfully requests the Applicants specify the conditions associated with each outgoing arrow. Furthermore, in blocks "Read Error?", "Parse Error?", and "Conversion Error?", each block misses one outgoing arrow and fails to indicate conditions (e.g., YES/NO) associated with each arrow;

For a sample correction (proposal only), "YES" of block "Read Input Scripts" should be deleted, the diamond shape of blocks "Read Error?", "Parse Error?", and "Conversion Error?" should be replaced by a rectangle shape, and the question marks "?" of these three blocks should be deleted;

Figure 7: acronym "AOM" should be spelled out to be more descriptive because the related text [0040-0044] does not mention about this block including "AOM"; and

Figure 11: "381" should be - -318- -.

Appropriate correction is required.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

9. The specification is objected to because of minor informalities. In [0046], line 4, "Logic 126" should be - -Logic 124- -.

Appropriate correction is required.

Claim Objections

10. Claims 1 and 6 is objected to because of minor informalities:

Claim 1, line 5, the phrase is considered to read as - -using semantic analysis to convert [[converting]] ...- -; and

Claim 6, "XML" should be spelled out at the first appearance in claims.

Appropriate correction is required.

Claim Rejections – 35 USC § 101

11. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

12. As set forth in the previous Office Action mailed October 6, 2006 (pp. 4-5), claim 49 is directed to non-statutory subject matter: "A carrier medium, comprising: ...".

Under the principles of compact prosecution, claim 49 has been examined as the Examiner anticipates the claim will be amended to obviate these 35 USC §101 rejection. For example, - -A memory, coupled to a processor, storing: ...- - as disclosed in the specification, [0054].

Claim Rejections – 35 USC § 102

13. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

14. Claims 1, 38, and 43 are rejected under 35 U.S.C. 102(e) as being anticipated by "WinRunner 7.0 Tutorial", published in 2000 (art made of record, hereinafter "WinRunner").

Claim 1:

WinRunner discloses *a method for transforming test cases that are converted to an abstract representation and storing abstract representation of test cases into a database system, comprising:*

importing test cases written in one or more scripting languages (e.g., pp. 34-38, Recording Tests; page 40, Stop Recording and Save);

using semantic analysis to convert test cases to an abstract representation that includes application state, wherein the application state is a set of application objects associated with a set of attributes and their values (e.g., pp. 42-43, GUI objects with attributes and values; pp. 22-26, GUI map with objects, attributes, and values),

external interaction sequences and input data (e.g., page 35, Choosing a record mode; pp. 38-40, Recording a context sensitive test; pp. 44-46, Recording in analog mode; pp. 11-12, performing operations on applications under tests),

or represents a runtime snapshot of an application under test which defines a context of external interaction (e.g., pp. 48-50, Running the test including sequences of actions, which provides a runtime snapshot of application under test); *and*

storing abstract representation of test cases into a database system (e.g., pp. 51-55, Test results; page 184, Maintain your test scripts; page 166, Test result window with test results; pp. 84-86, Running an existing test on a new version of an application under test; pp. 11-12, reusable tests).

Claim 38:

WinRunner discloses *a system for transforming test cases to an abstract representation that are stored in a database, comprising:*

a processor for importing test cases written in one or more scripting languages (e.g., pp. 34-38, Recording Tests; page 40, Stop Recording and Save);

logic that includes semantic analysis for converting test cases to an abstract representation that includes application state (e.g., page 35, Choosing a record mode; pp. 38-40, Recording a context sensitive test; pp. 44-46, Recording in analog mode; pp. 11-12),

external interaction sequences and input data (e.g., pp. 42-43, GUI objects with attributes and values; pp. 22-26, GUI map with objects, attributes, and values); *and*

a database that stores abstract representation of test cases, the application state being a set of application objects associated with a set of attributes and their values (e.g., pp. 51-55, Test results; page 184, Maintain your test scripts; page 166, Test result

window with test results; pp. 84-86, Running an existing test on a new version of an application under test),

or represents a runtime snapshot of an application under test which defines a context of external interaction (e.g., pp. 48-50, Running the test including sequences of actions, which provides a runtime snapshot of application under test).

Claim 43:

WinRunner discloses a *computer system for storing abstract representations of test cases in a database, comprising:*

a processor; a memory coupled to the processor (e.g., pp. 142-144),

the memory storing program instructions executable by the processor for converting test cases to an abstract representation that includes application state (e.g., pp. 34-38, Recording Tests; page 40, Stop Recording and Save; pp. 11-12),

external interaction sequences and input data (e.g., pp. 42-43, GUI objects with attributes and values; pp. 22-26, GUI map with objects, attributes, and values),

the application state being a set of application objects associated with a set of attributes and their values (e.g., page 35, Choosing a record mode; pp. 38-40, Recording a context sensitive test; pp. 44-46, Recording in analog mode),

or represents a runtime snapshot of an application under test which defines a context of external interaction (e.g., pp. 48-50, Running the test including sequences of actions, which provides a runtime snapshot of application under test)*; and*

a database that stores abstract representation of test cases (e.g., pp. 51-55, Test results; page 184, Maintain your test scripts; page 166, Test result window with test results; pp. 84-86, Running an existing test on a new version of an application under test).

Claim Rejections – 35 USC § 103

15. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior

art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

16. Claims 1-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent Publication No. 2004/0107415 A1 to Melamed et al. (art made of record, hereinafter "Melamed") in view of WinRunner.

Claim 1:

Melamed discloses *a method for transforming test cases that are converted to an abstract representation and storing abstract representation of test cases into a database system, comprising:*

importing test cases written in one or more scripting languages (e.g., [0009], [0013], [0054]; FIG. 13, Select test case, [0083]);

using semantic analysis to convert test cases to an abstract representation that includes application state (e.g., [0014], [0018], FIG. 14, [0093]),

external interaction sequences and input data (e.g., [0065], [0079], [0097]),

wherein the application state is a set of application objects associated with a set of attributes and their values (e.g., [0014], [0018], FIG. 14, [0093]),

or represents a runtime snapshot of an application under test which defines a context of external interaction (e.g., FIG. 18, blocks 188-189, 1810-1811, [0085-0086]); *and*

storing results *of test cases into a database system* (e.g., [0021], [0010], FIG. 18, [0040]).

Melamed discloses using WinRunner ([0010], [0065], [0079]), converting test cases to an abstract representation, and storing results of test cases into server 180 as set forth above, but does not explicitly disclose *storing abstract representation of test cases into a database system.*

However, WinRunner further discloses *storing abstract representation of test cases into a database system* (e.g., pp. 51-55, Test results; page 184, Maintain your test scripts;

page 166, Test result window with test results; pp. 84-86, Running an existing test on a new version of an application under test; pp. 11-12).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine WinRunner's teaching into Melamed's teaching. One would have been motivated to do so to run the existing tests on each new build, re-perform the same operations each time they are run, and test each build under repeated execution as suggested by WinRunner (e.g., pp. 11-12).

Claim 2:

The rejection of claim 1 is incorporated. Melamed also discloses *an application state represents a runtime snapshot of application under test, which defines the context of external interaction* (e.g., FIG. 18, [0085-0086]).

Claim 3:

The rejection of claim 2 is incorporated. Melamed also discloses *the application state includes a set of application objects, its attributes and attribute values* (e.g., [0014], [0018], [0093]).

Claim 4:

The rejection of claim 2 is incorporated. Melamed also discloses *the applications states corresponding to a test case are arranged in a hierarchical manner* (e.g., [0009], [0013], [0054]).

Claim 5:

The rejection of claim 2 is incorporated. Melamed also discloses *the database system is a relational database management system* (e.g., [0021], [0010], [0040]).

Claim 6:

The rejection of claim 2 is incorporated. Melamed also discloses *the database system is an XML database management system* (e.g., [0016], [0071]).

Claim 7:

The rejection of claim 2 is incorporated. Melamed also discloses *the scripting languages can be typed or untyped programming languages used for recording or authoring test cases* (e.g., [0090-0091]).

Claim 8:

The rejection of claim 2 is incorporated. Melamed also discloses *the external interaction sequences represent events invoked by external agents on the application objects* (e.g., FIG. 6, [0093]).

Claim 9:

The rejection of claim 8 is incorporated. Melamed also discloses *the external agents can be either human agents or other software agents* (e.g., [0094]).

Claim 10:

The rejection of claim 8 is incorporated. Melamed also discloses *the interaction sequencing includes flow control structures for capturing sequential, concurrent, looping and conditional interactions* (e.g., [0065], [0079]).

Claim 11:

The rejection of claim 2 is incorporated. Melamed also discloses *implementing a syntax analyzer for incoming scripts* (e.g., [0051-0053]).

Claim 12:

The rejection of claim 11 is incorporated. Melamed also discloses *the syntax analyzer is implemented one for each scripting language* (e.g., [0052-0053]).

Claim 13:

The rejection of claim 12 is incorporated. Melamed also discloses *the syntax analyzer utilizes rules of syntax analysis that are specified in Extended Backus-Naur Form (EBNF)* (e.g., [0016], [0051]).

Claim 14:

The rejection of claim 12 is incorporated. Melamed also discloses *the syntax analyzer generates a parse tree in the form of an Abstract Syntax Tree (AST)* (e.g., 0054], [0076]).

Claim 15:

The rejection of claim 2 is incorporated. Melamed also discloses *implementing a semantic analysis that converts a abstract syntax tree to an abstract test case representation based on an Application Object Model (AOM)* (e.g., [0016], [0051]).

Claim 16:

The rejection of claim 15 is incorporated. Melamed also discloses *the semantic analysis decomposes the test cases represented as an Abstract Syntax Tree into application state, external interaction sequences and input data* (e.g., [0072-0074]).

Claim 17:

The rejection of claim 15 is incorporated. Melamed also discloses *an application object model is a metadata representation for modeling application under test* (e.g., [0016], [0051]).

Claim 18:

The rejection of claim 17 is incorporated. Melamed also discloses *the metadata representation includes object type definitions for application objects* (e.g., [0054], [0076]).

Claim 19:

The rejection of claim 17 is incorporated. Melamed also discloses *the metadata representation includes attribute definitions for each application object type* (e.g., [0016], [0071]).

Claim 20:

The rejection of claim 17 is incorporated. Melamed also discloses *the metadata representation includes definition of methods and events that are supported by each application object type* (e.g., [0014], [0018]).

Claim 21:

The rejection of claim 17 is incorporated. Melamed also discloses *the metadata representation includes definition of effects of events on an application state* (e.g., [0093], [0097]).

Claim 22:

The rejection of claim 18 is incorporated. Melamed also discloses *application object type definitions include additional categorization of each application object types into hierarchical, container and simple types* (e.g., [0093], [0014], [0018]).

Claim 23:

The rejection of claim 22 is incorporated. Melamed also discloses *the hierarchical object types are associated with an application state of its own; wherein application object types that can contain instances of other objects are termed as container types* (e.g., [0021], [0010], [0040]).

Claim 24:

The rejection of claim 23 is incorporated. Melamed also discloses *the state associated with a hierarchical application object type is a modal application state or a nonmodal application state* (e.g., [0009], [0013], [0054]).

Claim 25:

The rejection of claim 24 is incorporated. Melamed also discloses *a modal application state restricts possible interactions to application object instances available within the current application state* (e.g., [0085-0086]).

Claim 26:

The rejection of claim 22 is incorporated. Melamed also discloses *the effects of events on an application state capture one or more consequences of the event to the application state* (e.g., [0065], [0079]).

Claim 27:

The rejection of claim 26 is incorporated. Melamed also discloses *a consequence of an event is selected from, creation of a new object instance of a given type, deletion of an object instance of a given type, modification of attributes of an existing object instance and selection of an instance of an object type* (e.g., [0093], [0018]).

Claim 28:

The rejection of claim 27 is incorporated. Melamed also discloses *creation of a new instance of an object of type that is hierarchical results in creation of a new application state* (e.g., [0054-0056]).

Claim 29:

The rejection of claim 27 is incorporated. Melamed also discloses *selection of an object instance of type that is hierarchical results in selection of the application state associated with that object instance* (e.g., [0065], [0079]).

Claim 30:

The rejection of claim 2 is incorporated. Melamed also discloses *enriching the abstract representation of test cases with information from an application metadata repository* (e.g., [0021], [0010]).

Claim 31:

The rejection of claim 30 is incorporated. Melamed also discloses *the enrichment of abstraction representation of test cases involves extracting values for those attributes of application objects associated with the test cases that are missing in the incoming test scripts* (e.g., [0054], [0083]).

Claim 32:

The rejection of claim 30 is incorporated. Melamed also discloses *enriching the abstraction representation of test cases includes decoupling of test cases from their recording or authoring environments* (e.g., [0093-0095]).

Claim 33:

The rejection of claim 30 is incorporated. Melamed also discloses *enriching the abstraction representation of test cases allows usage of attributes that are stable within an application metadata representation* (e.g., [0054], [0076]).

Claim 34:

The rejection of claim 33 is incorporated. Melamed also discloses *using an identification field for a given object within the application metadata repository improves the reusability of a test case instead of a label used to represent the same object within a user interface which can change based on the locale of the application* (e.g., [0072-0074]).

Claim 35:

The rejection of claim 33 is incorporated. Melamed also discloses *using an identification field allows to overcome the problem of different test execution environments using different attributes to identify the same application object* (e.g., [0016], [0051]).

Claim 36:

The rejection of claim 35 is incorporated. Melamed also discloses *enriching the abstraction representation of test cases enables representation of test cases that are test execution environment independent* (e.g., [0013], [0054]).

Claim 37:

The rejection of claim 2 is incorporated. Melamed also discloses *separating application object attributes and input data from external interaction sequencing provides automatic parameterization* (e.g., [0054], [0076]).

Claim 38:

Melamed discloses *a system for transforming test cases to an abstract representation that are stored in a database, comprising:*

a processor for importing test cases written in one or more scripting languages (e.g., [0009], [0013], [0054]; FIG. 13, Select test case, [0083]);

logic that includes semantic analysis for converting test cases to an abstract representation that includes application state (e.g., [0014], [0018], FIG. 14, [0093]),

external interaction sequences and input data (e.g., [0065], [0079], [0097]); *and*

a database that stores results *of test cases, the application state being a set of application objects associated with a set of attributes and their values* (e.g., [0014], [0018], FIG. 14, [0093]),

or represents a runtime snapshot of an application under test which defines a context of external interaction (e.g., [0021], [0010], FIG. 18, [0040]).

Melamed discloses using WinRunner ([0010], [0065], [0079]), converting test cases to an abstract representation, and storing result of test cases into server 180 as set forth above, but does not explicitly disclose *storing abstract representation of test cases into a database system.*

However, WinRunner further discloses *storing abstract representation of test cases into a database system* (e.g., pp. 51-55, Test results; page 184, Maintain your test scripts; page 166, Test result window with test results; pp. 84-86, Running an existing test on a new version of an application under test).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine WinRunner's teaching into Melamed's teaching. One would have been motivated to do so to run the existing tests on each new build, re-perform the same operations each time they are run, and test each build under repeated execution as suggested by WinRunner (e.g., pp. 11-12).

Claim 39:

The rejection of claim 38 is incorporated. Melamed also discloses *a syntax analyzer for incoming scripts* (e.g., [0051-0053]).

Claim 40:

The rejection of claim 38 is incorporated. Melamed also discloses *logic for implementing a semantic analysis that converts the abstract syntax tree to an abstract test case representation based on an Application Object Model (AOM)* (e.g., [0016], [0071]).

Claim 41:

The rejection of claim 38 is incorporated. Melamed also discloses *logic for selecting an object instance of type that is hierarchical results in selection of the application state associated with that object instance* (e.g., [0065], [0079]).

Claim 42:

The rejection of claim 38 is incorporated. Melamed also discloses *logic for enriching the abstraction representation of test cases to enable representation of test cases that are test execution environment independent* (e.g., [0013], [0054]).

Claim 43:

Melamed discloses a *computer system for storing abstract representations of test cases in a database, comprising:*

a processor; a memory coupled to the processor (e.g., FIG. 18, [0040]),

the memory storing program instructions executable by the processor for converting test cases to an abstract representation that includes application state (e.g., [0009], [0013], [0054]; FIG. 13, Select test case, [0083]),

external interaction sequences and input data (e.g., [0065], [0079], [0097]),

the application state being a set of application objects associated with a set of attributes and their values (e.g., [0014], [0018], FIG. 14, [0093]),

or represents a runtime snapshot of an application under test which defines a context of external interaction (e.g., [0065], [0079], [0097]); *and*

a database that stores results *of test cases* (e.g., [0021], [0010], FIG. 18, [0040]).

Melamed discloses using WinRunner ([0010], [0065], [0079]), converting test cases to an abstract representation, and storing result of test cases into server 180 as set forth above, but does not explicitly disclose *a database that stores abstract representation of test cases.*

However, WinRunner further discloses *a database that stores abstract representation of test cases* (e.g., pp. 51-55, Test results; page 184, Maintain your test scripts; page 166, Test result window with test results; pp. 84-86, Running an existing test on a new version of an application under test).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine WinRunner's teaching into Melamed's teaching. One would have been motivated to do so to run the existing tests on each new build, re-perform the same operations each time they are run, and test each build under repeated execution as suggested by WinRunner (e.g., pp. 11-12).

Claim 44:

The rejection of claim 43 is incorporated. Melamed also discloses *a syntax analyzer for incoming scripts* (e.g., [0051-0053]).

Claim 45:

The rejection of claim 44 is incorporated. Melamed also discloses *the syntax analyzer generates a parse tree in the form of an Abstract Syntax Tree (AST)* (e.g., [0054], [0076]).

Claim 46:

The rejection of claim 45 is incorporated. Melamed also discloses *logic to implement semantic analysis and convert the AST to an abstract test case representation based on an Application Object Model (AOM)* (e.g., [0016], [0071]).

Claim 47:

The rejection of claim 43 is incorporated. Melamed also discloses *logic for enriching the abstract test case representation with information from an application metadata repository* (e.g., [0021], [0010]).

Claim 48:

The rejection of claim 43 is incorporated. Melamed also discloses *logic for separating application object attributes and input data from external interaction sequencing to provide automatic parameterization* (e.g., [0054], [0076]).

17. Claim 49 is rejected under 35 U.S.C. 103(a) as being obvious over Rosaria (art of record, US Patent No. 6,976,246).

Claim 49:

Rosaria discloses *converting test cases to an abstract representation that includes application state, external interaction sequences and input data* (e.g., FIG. 3, col.6: 18-65; FIG. 6, col.9: 16 – col.10: 56).

Rosaria further discloses storing data of test case models, intermediate and executable files of the test case generator program in volatile and nonvolatile memories of the system (e.g., FIG. 4, col.7: 32 – col.8: 46), saving graph traversal programs (e.g., graph.exe), model files (e.g., clock.std), test case files (e.g., clock.tsc) in database (col.6: 28-37 and col.14: 62 – col.15: 21). Rosaria also discloses a test oracle function can be called to verify results/states/status from the tests stored in the system (e.g., col.4: 16-62).

Thus, it would have been obvious to one having ordinary skill in the art of test case models to store *program instructions of abstract representations of test cases in a database* in order to provide better data organization and manageability for the program application.

Conclusion

18. Applicants' amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

19. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone is (571) 272 8570. The examiner can normally be reached on the first Monday of the bi-week, and every Tuesday, Thursday, and Friday from 6:00AM to 6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

T. Dao

TUAN DAM
SUPERVISORY PATENT EXAMINER